DOCUMENT RESUME

ED 309 192 TM 013 672

AUTHOR van der Linden, Wim J.; Boekkooi-Timminga, Ellen
TITLE A Maximin Model for Test Design with Practical Constraints. Project Psychometric Aspects of Item Banking No. 25. Research Report 87-10.
INSTITUTION Twente Univ., Enschede (Netherlands). Dept. of Education.
SPONS AGENCY Netherlands Inst. for the Advancement of Pure Research, The Hague.
PUB DATE Dec 87
NOTE 37p.
AVAILABLE FROM Mediatheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.
PUB TYPE Reports - Research/Technical (143)

EDRS PRICE MF01/PC02 Plus Postage.
DESCRIPTORS Algorithms; Foreign Countries; Item Banks; *Latent Trait Theory; *Linear Programing; Mathematical Models; *Test Construction; Testing Problems; Test Items
IDENTIFIERS *Constraints; *Maximin Model; Three Parameter Model

ABSTRACT

A "maximin" model for item response theory based test design is proposed. In this model only the relative shape of the target test information function is specified. It serves as a constraint subject to which a linear programming algorithm maximizes the information in the test. In the practice of test construction there may be several demands with respect to the properties of the test. The way in which these can be formulated as linear constraints in the model is demonstrated. The constraints discussed include: (1) test composition; (2) administration time; (3) selection of item features; (4) group-dependent item parameters; (5) inclusion or exclusion of individual items; and (6) inter-item dependencies. An example of a test construction problem with practical constraints is presented. Using the three-parameter logistic model, an item bank of 1,000 items was drawn for the application of the test construction model, which was solved using the computer program LINPROG. Some alternative models of test construction are discussed. Three tables provide information about four solutions and list alternative objective functions in test construction. (SLD)

****
* Reproductions supplied by EDRS are the best that can be made *
* from the original document. *
****

ED309192

A Maximin Model for Test Design with Practical Constraints

Research Report 87-10

U.S. DEPARTMENT OF EDUCATION
Office of Educational Research and Improvement
EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)

☑ This document has been reproduced as received from the person or organization originating it.

☐ Minor changes have been made to improve reproduction quality.

• Points of view or opinions stated in this document do not necessarily represent official OERI position or policy.

"PERMISSION TO REPRODUCE THIS MATERIAL HAS BEEN GRANTED BY

J. NELISSEN

TO THE EDUCATIONAL RESOURCES INFORMATION CENTER (ERIC)."

Wim J. van der Linden
Ellen Boekkooi-Timminga

department of EDUCATION

University of Twente

Division of Educational Measurement and Data Analysis

TM013612

Colofon:
Typing: Mevr. L.A.M. Padberg
Cover design: Audiovisuele Sectie TOLAB Toegepaste Onderwijskunde
Printed by: Centrale Reproductie-afdeling

A Maximin Model for Test Design with Practical Constraints

W.J. van der Linden

E. Boekkooi-Timminga

To appear in Psychometrika

A Maximin Model for Test Design with Practical Constraints / Wim J. van der Linden & Ellen Boekkooi-Timminga - Enschede : University of Twente, Department of Education, December, 1987. - p.

Abstract

A maximin model for IRT-based test design is proposed. In the model only the relative shape of the target test information function is specified. It serves as a constraint subject to which a linear programming algorithm maximizes the information in the test. In the practice of test construction, several demands with respect to the properties of the test may exist. The paper shows how these can be formulated as linear constraints in the model. A worked example of a test construction problem with practical constraints is presented. The paper concludes with a discussion of some alternative models of test construction.

Keywords: Item Response Theory; Test Construction; Linear Programming

A Maximin Model for Test Design with Practical Constraints

In item response theory (IRT), test design is usually based on the concepts of item and test information functions. In this paper, as an example, information functions under the three-parameter logistic model for dichotomous responses are considered. The results, however, hold for any IRT model.

The three-parameter logistic model is as follows:

$$(1) \qquad p_i(\theta) = c_i + (1-c_i)\,(1+\exp[-a_i(\theta-b_i)])^{-1} \ .$$

where $\theta \in \langle -\infty, +\infty \rangle$ is the ability measured by the test items, $a_i \in [0, +\infty\rangle$ and $b_i \in \langle -\infty, +\infty \rangle$ are parameters for the discriminating power and difficulty of item i, and $c_i \in [0,1]$ is the probability of solving item i for $\theta \to -\infty$. The model gives the probability of a correct response as a function of the ability parameter θ. For known item parameters, it holds that Fisher's information about the unknown θ in a single response, U_i, $(u_i=0, 1)$, to item i is equal to

$$(2) \qquad I(U_i;\theta) = p_i'(\theta)/\{p_i(\theta)[1-p_i(\theta)]\}^2 \ .$$

For a sample of locally independent responses, $U_1, \ldots, U_I$, to the items $i = 1, \ldots, I$, Fisher's information is equal to

$$(3) \qquad I(U_1,\ldots,U_I;\theta) = \sum_{i=1}^{I} p_i'(\theta)/(p_i(\theta)[1-p_i(\theta)])^2 \ .$$

For item parameters estimated from response data with sufficient precision, Birnbaum (1968) introduced (2) and (3) as the item and test information functions.

The additivity in (3) suggests the following procedure for test construction: A target information function for the test is specified. Items are selected to fill the area under the target function. The procedure is stopped as soon as the sum of the item information functions exceeds the target. In Birnbaum's (1968) and Lord's (1980) description of the procedure it is assumed that the selection is done by hand. In general, however, finding an optimal solution (e.g., a test of minimal length exceeding the target) by hand is practically impossible. Even an approximate solution may involve several cycles of back-tracking. Also, specifying a target information function is not an easy task. Although a test constructor may be able to provide the desired shape of the curve (e.g., a flat curve for a diagnostic test or a peaked one for decision making), the necessity to decide on its exact heigth is likely to create a problem, the reason being that the metric of the information measure has no meaning to the average test constructor.

Recently, a series of papers has been published in which the Birnbaum-Lord procedure is replaced by an algorithm from zero-one programming (Rao, 1985; Wagner, 1975). The idea to

apply zero-one programming to test construction was already suggested in Yen (1983). Theunissen (1985) was the first to present a zero-one programming model for test construction with a target information function. In the model, the number of items in the test is the objective function to be minimized subject to the (linear) conditions that, at a number of θ-values the information in the test is above the target. The same idea has been explored in Boekkooi-Timminga (1987), Boekkooi-Timminga and van der Linden (1987), Theunissen (1986), Theunissen and Verstralen (1986), and van der Linden and Boekkooi-Timminga (1988). In all these papers, it is still assumed that the test constructor is able to specify the exact heigth of the target information function at a number of points. For a procedure enabling the test constructor to do so, see Kelderman (1987).

It is the purpose of this paper to present a maximin model for test construction. In this model only the relative shape of the target test information function has to be provided. A simple experiment to elicit this shape from a test constructor is described in the next section. The data from the experiment are then used to specify a linear constraint in a model that maximizes the information in the test. Since the objective function in the model does not contain any item or test parameter, all properties of the test can be controlled by including additional constraints in the model. In the practice of test construction, several demands with respect to the properties of the test may exist

– e.g., with respect to the composition of the test, the administration time, the curricular fit, and possible links between the contents of the items. It is another purpose of this paper to show how such demands can be formulated as linear constraints in the decision variables. A worked example of the model including several of these constraints is given. The paper concludes with a discussion of some alternative zero–one programming models of test construction.

A Maximin Model

Instead of considering target information functions over the whole range of θ–values zero–one programming models only assume target values at certain points. One reason for this is that item information functions are continuous, well–behaved functions for which the value of the sum at a certain point does not differ drastically from those at neighboring points. Another, more practical motivation is that interest often exist only at certain critical ability levels ignoring the properties of the test at other levels, for example, when the test is to be used for decision making. Hence, in this paper it is also assumed that a discrete approach is appropriate, provided the number of points and their positions are free.

The following experiment is proposed to elicit the relative shape of the target information function from the test constructor. First, the test constructor is faced with

the ability scale underlying the item bank. This can be done by offering him or her a line displaying the contents of items with locations at some well-chosen points. The same practice is used in scale-score reporting of assessment data (e.g., Pandey, 1986). Then, the constructor is asked to select a number of scale points he or she wants to consider. The number of points and their spacing are free. Let θ_k, $k = 1, \ldots, K$, denote these points. Next, he or she is given a fixed number of chips (100, say) and requested to distribute them over the scale points such that they reflect the relative distribution of information wanted from the test. The final step then is to ask the test constructor for the desired number of items in the test. The answer to this question can be facilitated by providing some statistics about the time typically needed by the group of examinees to complete items in the bank.

The Model

Now the idea is to select the items such that they maximize the information in the test, while the resulting test information function still has the desired shape. Let r_k be the numbers of chips the test constructor puts at point θ_k ($k = 1,\ldots,K$).The relative target information function is characterized by a series of lower bounds ($r_1y, \ldots, r_Ky$) in which y is a dummy variable to be maximized subject to the constraint that test length is equal tothe value n specified by the test constructor. Finally, x_i ($i = 1,\ldots,I$) is the

decision variable as to whether (x_i=1) or not (x_i=0) to include item i in the test. This leads to the following model:

(4) maximize y

subject to

$$(5) \quad \sum_{i=1}^{I} I_i(\theta_k)x_i - r_k y \geq 0 \qquad k = 1, \ldots, K$$

$$(6) \quad \sum_{i=1}^{I} x_i = n$$

$$(7) \quad x_i \in \{0, 1\} \qquad i = 1, \ldots, I.$$

$$(8) \quad y \geq 0$$

The constraints in (5) set a series of lower bounds, $r_k y$, to the test information $I_t(\theta_k) = \sum_{i=1}^{I} I_i(\theta_k)x_i$ at each of the points θ_k. The common factor y in these bounds is maximized in (4). The constraint in (6) sets the test length equal to n.

If the left-hand side of the restrictions in (5) were divided by r_k the model would have new coefficients $I_i(\theta_k)r_k^{-1}$ for the decision variables x_i and a coefficient equal to one for variable y. In this representation it is clear that y can be considered a lower bound to the weighted sums of decision

variables $\sum_{i=1}^{I} I_i(\theta_k) r_k^{-1} x_i$ and that the values of x_i are selected such that this lower bound is maximal. Hence, mathematically the model is of the maximin type. To solve the model for the values of x_i, $i = 1, \ldots, I$, and y, a branch-and-bound algorithm from integer programming can be used (e.g., Wagner, chap. 13). Such algorithms are readily available in computer code nowadays.

Some Practical Constraints

For algorithmic test design to be practical, it is necessary to provide control of features of the test other than just the information function and the number of items. It should be noted that the objective function in (4) is a dummy variable introduced to cast the maximin criterion into a linear model. It does not contain any item or test parameters, and therefore does not explicitly control the values of these parameters. For this purpose, however, additional constraints can be included in the model. In this section a review of constraints to be met in the practice of test construction is given, and it is shown how these can be modeled into a linear form. Throughout this section it is assumed that (4) through
(8) is the basic model.

Test Composition

As already noted, for a sufficiently large bank of test items, the constraint in (6) controls the length of the test. The same principle can be applied at the level of possible subtests providing the test constructor with the ability to control the composition of the test. Let V_j ($j = 1,\ldots,J$) be a subset of items in the bank from which the test constructor wants $n_j \leq n$ in the test. This is attained if the following equality is added to the model:

$$(9) \qquad \sum_{i \in V_j} x_i = n_j \qquad j = 1, \ldots, J$$

It is important to note that using a series of such constraints provides the opportunity for controlling the composition of the test simultaneously with respect to several dimensions. For example, an item bank for English could be partitoned not only with respect to its content (e.g., vocabulary, grammar, or reading comprehension), but also to a behavioral dimension (e.g., knowledge of facts, application of rules, or evaluation) or the format of its items (e.g., multiple choice, completion, or matching). For each set in these partitions the constraint in (9) is incorporated within the model, with the restriction that the n_j's are specified such that the sum over all sets in the same partition is equal to n. If this option is used, the constraint in (6) is redundant and may be dropped. Finally,

observe how this example shows that (9) can be used with repect to both disjoint and non-disjoint subsets of items.

Administration Time

In a computerized testing environment, the time needed to solve the items in the bank by the population of examinees of interest can easily be monitored. Let t_i be, e.g., the 95th percentile of the distribution of time for item i in the population. Instead of fixing the length of the test, the selection of the items could also be based on the time limit, T, in force for the examinees. In that case (6) can be replaced by

$$(10) \qquad \sum_{i=1}^{I} t_i x_i \leq T.$$

However, if there is a reason to restrict the number of items in the test as well, (10) can also be used in combination with (6) replacing the equality in the latter by an inequality.

Analogous to (9), the composition of the test can be controlled by introducing time limits at the subtest level.

Selection on Item Features

Including the constraints below in the model, it is possible to give all items in the test the same feature.

Let c_i be a positively valued numerical parameter representing a feature of the items in the bank. Then it is

possible to restrict the selection of the items to those with $c_i \in [c_\ell, c_u]$ by including the following set of inequalities in the model:

$$c_i x_i \leq c_u \qquad i = 1, \ldots, I \tag{11}$$

$$c_i^{-1} x_i \leq c_\ell^{-1} \qquad i = 1, \ldots, I, \tag{12}$$

where $c_u > c_\ell$.

Unlike (9), these constraints do not fix the length of subtests. They are used to give all items in the test the same properties. At the same time, (9) can be used to compose the test with different item properties.

If the frequency of administration of the items in the bank is monitored, the constraints in (11) through (12) can be used to restrict the selection of the items to certain frequencies. For example, if the intention is to obtain uniform usage of items in the bank, (11) can be used to set an upper bound for item use thus restricting the selection of items to those with lower usage.

Another example of the use of (11) through (12) is to restrict the administration time, t_i, for each individual item in the test to certain limits.

It is also possible to substitute one of the parameters in the item response model for c_i. In this way, the constraints can be used, for example, to select items with values for the difficulty parameters in a certain interval.

For the Rasch model, this allows for the selection of items based on their probabilities of success: Let θ_o be the a priori known average ability of the group of examinees, and let $[p_\ell, p_u]$ be the interval to which the probabilities of success for the "average" examinees are restricted. It follows that the items must have the values of the difficulty parameter, b_i, in the interval $[b_\ell, b_u]$ determined by $p(\theta_o; b_\ell) = p_u$ and $p(\theta_o; b_u) = p_\ell$, where $p(.)$ is the logistic function specified in the Rasch model. Selecting items based on their probabilities of success for given examinees may be desirable for instructional reasons. Constraints like (11) – (12) need not enter the optimization phase of the procedure. They imply that certain items, and hence their decision variables, are excluded from the model. Normally, a reduction phase precedes the actual optimization in which such constraints are used to give the model its most economical form.

Group-dependent Item Parameters

If the item bank has to serve distinct groups of examinees, items may have different properties for different groups. In such cases it is obvious to consider the parameter c_i in (11) – (12) as group dependent. In school settings, for instance, the recording of the date of the final administration of item i to group $g = 1, \ldots, G$ may be useful. The constraint in (12), with c_{gi} instead of c_g, then allows the selection of items for one group that have not been used after a given

date for other groups. Such strategies may be instrumental in solving the problem of test security.

If c_{gi} is allowed to take only the values zero and one, it can be used to adapt tests to curriculum differences between groups. Let c_{gi} indicate whether (c_{gi}=1) or not (c_{gi}=0) item i covers a part of the curriculum of group g. Then the following constraint automatically suppresses the administration of items to group g on topics for which instruction is absent:

$$x_i \leq c_{gi} \qquad i = 1, \ldots, I. \tag{13}$$

Inclusion or Exclusion of Individual Items

For some personal reason the test constructor may want to include or exclude certain items from the test. As already noted by Theunissen (1985) and Boekkooi-Timminga (1987), the following constraints can be used for this purpose:

$$x_i = 0 \qquad i \in V_j^{(1)} \tag{14}$$

$$x_i = 1 \qquad i \in V_j^{(0)} \tag{15}$$

with

$$V_j^{(0)} \cap V_j^{(1)} = \emptyset$$

where $V_j^{(0)}$ and $V_j^{(1)}$ are subsets of items from the bank from which all or none of the item should be included in the test, respectively.

Inter-item Dependencies

In some practical situations certain items are not allowed to be administered on the same test. For instance, this may be the case if some items contain a cue with respect to the solution of other items. Suppose V_j ($j = 1, \ldots, J$) now indicates a subset of mutually exclusive items in the bank. Then, the following multiple-choice constraint allows the selection of at most one item from this set:

$$(16) \qquad \sum_{i \in V_j} x_i \leq 1 \qquad j = 1, \ldots, J$$

By raising the right-hand side of this constraint more items are allowed to be selected.

The opposite case occurs if the selection of one item entails the necessity to select other items as well. This may occur if the content of some items builds on that of other items. (The question if one of the current response models could fit such items is deliberately omitted.) It is also possible to model the presence of this dependency between test items as a linear constraint. Let V_j ($j = 1, \ldots, J$) now represent a subset of items in the bank of which the selection of any item impkies the selection of all. The

following equality guarantees the simultaneous inclusion or exclusion of these items from the test:

$$(17) \qquad \sum_{i \in V_j} x_i = |V_j| x_{i_j}$$

where $|V_j|$ denotes the number of items in V_j and i_j is an arbitrary item in V_j.

Another approach to the problem of inter-item dependencies, using Boolean algebra, is given by Theunissen (1986).

Computational Procedures

It is well known that 0-1 programming is NP-hard. For the present test construction problem this means that if the number of items in the bank (and hence of the decision variables in the model) increases, the amount of CPU time needed for an exact solution may become unpractical. In such situations approximate solutions with good properties are required.

In general two approaches have been followed by the authors to arrive at approximate solutions: (1) relaxing the assumption of discreteness of the decision variable in the model and rounding the result, and (2) stopping a branch-and-bound algorithm after a number of steps. If the assumption of discreteness is relaxed, the decision variables are allowed to take values in the interval <0,1> and the problem boils

down to a regular linear programming model which can be solved quickly using the simplex algorithm. A well-known theorem in mathematical programming guarantees that the number of fractional values for the decision variables is maximally equal to the number of constraints in the model (Dantzig, 1957). However, practical experience by the authors and others in the field has shown that mostly this number is much lower than its maximum. For the test construction model in this paper, it is not possible to round all fractional values to one, as is done, e.g., in Theunissen's (1985) model of minimal test lenght, because then constraint (6) is not always met. A more favorable, therefore, is optimal rounding. In this option, the constraints in the model are adapted to the characteristics of the items already selected in the test with decision variable values one, and next an exact solution is computed for the remaining 0-1 problem in which only the few decision variables with fractional values in the original solution are considered. The adaptation of the constraints is a simple task which can easily be implemented in computer code. For example, in the optimal rounding procedure constraint (5) has to be replaced by

$$(18) \qquad \sum_{f=1}^{F} I_f(\theta_k)x_f - r_k y + I^*(\theta_k) \geq 0 \qquad k = 1, \ldots, K$$

where $f = 1, \ldots, F$ are the items having fractional values in the original solution, $I_f(\theta_k)$ is their information at θ_k, and $I^*(\theta_k)$ is the sum of the information of the items with

$x_i = 1$. Because the number of fractional values is small, the 0–1 problem can be solved quickly and the value of the objective function will differ hardly, if at all, from its optimum. Hence, optimal rounding will give excellent results.Optimal rounding does not produce results necessarily meeting all types of constraints, though. This can be shown by a simple example: Consider the constraint $x_1 - x_2 = 0$, and, assume that the solution to the relaxed model contains $x_1 = x_2 = 0.5$. Then, the 0–1 problem will not have a solution, because both or none of the variables should take the value 1. Fortunately, such cases can directly be seen from the solution to the relaxed model. If not, it only takes a very small amount of CPU–time to detect the problem. Then the following approach to solving the 0–1 model can be applied.

In the second approach, a branch–and–bound algorithm is used to solve the original 0–1 problem stopping the process after a number of steps, e.g. after the first 0–1 solution is found. This solution also fits the constraints but experience has shown that the objective function value generally is not so close to its optimum as in the previous approach.

Finally, it is stressed again that not all of the constraints in the model need to enter the optimization process. Normally, a reduction of the model precedes the actual optimization. Some of the specifications by the test constructor result in fixed variables, whereas others may entail redundant constraints. In the reduction phase, all

fixed variables are removed from the model adapting the remaining constraints when necessary. Next, redundant constraints are traced and eliminated. This procedure was followed in the below example.

Example

Using the three-parameter logisitic model, an item bank of 1,000 items was randomly and independently drawn from the following distribution: a_i ~ U(0.5;1.5), b_i ~ N(0;1), and c_i ~ U(0;0.4). Furthermore, the items were given administration times (in minutes) from t_i ~ U(3;11), two groups of examinees with different but overlapping curricula were considered giving the items curricular fit indices c_{gi} = 1 with probability .50 for each group, and 50%, 25% and 25% of the items were considered to be of the multiple-choice, matching, and completion type, respectively.

A test was designed with equal information at θ = -2, 0, and 2. Thus r_k was set equal to one for all k. Also, the following constraints applied: (1) No items with discrimination parameters a_i < 1.0 or guessing parameters c_i > 0.2 were allowed to enter the test; (2) The test should fit the curricula of both groups of examinees; (3) The maximum administration time for the test was equal to 100 minutes; (4) Items 251 and 256 were not allowed to enter the test, but items 300 – 301 had to be in the test; (5) The test

should consist of 10 multiple choice, 5 matching, and 5 completion items.

The test construction model was solved using the program LINPROG (Anthonisse, 1984) on an Olivetti M24 personal computer with hard disk and mathematical coprocessor. Four solutions were obtained: (1) the optimal 0-1 solution (branch-and-bound); (2) the solution to the relaxed model (simplex); (3) the optimally rounded solution (simplex and next branch-and-bound); and (4) the first 0-1 solution (truncated branch-and-bound). Solution 3 and 4 were obtained using the procedures described above. In Table 1, the CPU times and the numbers of iterations of the solutions are

---

Insert Table 1 about here

---

given. As could be expected, the time needed for the exact 0-1 solution, albeit not prohibitive for some personal computer applications, is much larger than for the other solutions. Optimally rounding the simplex solution, which had six fractional values, required only 3.60 seconds. Extensive simulation studies of test construction models (Timminga, 1985) have shown that the CPU-time tends to increase with the numbers of θ-points, items and constraints in the model. No systematic effects could be observed for the spacing of the

θ-points and the distributions of the parameter values of the items in the bank.

Table 2 summarizes the characteristics of the tests

Insert Table 2 about here

---

selected in the four solutions. The test for the simplex solution was obtained by rounding to the nearest integer. As a consequence, this solution does not meet the constraint on administration time and demands a little more than T = 100 minutes. Like the other solutions, this solution does meet all other constraints. Table 2 also shows that the optimally rounded simplex solution coincides with the optimal 0-1 solution. This will often take place with few fractional values in the solution, but is not necessarily so.

Discussion

In this paper an IRT-based test construction model with a maximin objective function was proposed. Instead of a maximin function, other objective functions could be used for test construction. Table 3 gives a list of possible alternatives

Insert Table 3 about here

considered earlier in the literature or studied by the authors. Except for objective function 4, for each function in the list it is assumed that in addition the following set of target reachability constraints is included in the model

$$\sum_{i=1}^{I} I_i(\theta_k)x_i \geq I(\theta_k), \qquad k = 1, \ldots, K, \tag{19}$$

where $I(\theta_k)$ denotes the value of the target information function at θ_k. Compared with the objective functions in Table 3, the maximin function combines two favorable features. First, it is the only one that has the advantage of a relatively easy way of specifying the target information function. Unlike the other models, no familiarity whatsoever with the metric of the information measure is assumed. Two, the basic model in (4) through (8) always has a feasible solution for test lengths $n \leq I$. Other models may have feasibility problems when, along with the required test length, an absolute target information function has to be specified. (Objective function 1 solves this problem in a different way).

Strictly speaking, a solution to a test construction model is just a collection of test items. To make them into a test, the items should be put into an appropriate order of

administration. This again can be considered as a problem of optimization subject to constraints with respect to, e.g., item difficulty, administration time, or topic structure. How this problem can be solved using a linear programming model is the subject of another paper.

As a final comment it is noted that in a computerized test system test construction models can also be used in an interactive mode. In doing so, the system selects a test and requests the user to indicate which items are appropriate and which are not. In the next stage, the model is used to select a new version of the test, but now with $x_i = 1$ for the items that have to be retained and $x_i = 0$ for those that were rejected. The process is repeated until all items are considered appropriate. Interactive use of test construction models is recommended since it allows test construction to be based on possible remaining constraints of interest that cannot be modeled as linear (in)equalities.

References

Anthonisse, J.M. (1984). Linprog. Amsterdam: Centre for Mathematics and Computer Science CWI.

Birnbaum, A. (1968). Some latent trait models and their use in inferring an examinee's ability. In F.M. Lord & M.R. Novick, Statistical theories of mental test scores. Reading, Mass.: Addison-Wesley.

Boekkooi-Timminga, E. (1987). Simultaneous test construction by zero-one programming. Methodika, 1, to appear.

Boekkooi-Timminga, E., & van der Linden, W.J. (1987). Algorithms for automated test construction. In F.J. Maarse, L.J.M. Mulder, W.P.B. Sjouw, & A.E. Akkerman, Computers in psychology: Methods, instrumentation and psychodiagnostics. Lisse: Swets & Zeitlinger.

Dantzig, G. (1957). Discrete-variable extremum problems. Operations Research, 5, 266-277.

Kelderman, H. (1987). Some procedures to assess target information functions. In W.J. van der Linden (Ed.), IRT-based test construction (Research Report 87-2). Enschede: University of Twente, Department of Education

Lord, F.M. (1980). Applications of item response theory to practical testing problems. Hillsdale, N.J.: Lawrence Erlbaum.

Pandey, T.N. (1986). State of the art of large-scale assessment in the United States. In W.J. van der Linden & J.M. Wijnstra (Eds.), Ontwikkelingen in de methodologie van het onderwijs onderzoek [Developments in the methodology of educational research]. Lisse: Swets & Zeitlinger.

Rao, S.S. (1985). Optimization: Theory and applications (2nd Ed.). New Dehli: Wiley Eastern Ltd.

Theunissen, T.J.J.M. (1985). Binary programming and test design. Psychometrika, 50, 411-420.

Theunissen, T.J.J.M. (1986). Optimization algorithms in test design. Applied Psychological Measurement, 10, 381-389.

Theunissen, T.J.J.M., & Verstralen, H.H.F.M. (1986). Algoritmen voor het samenstellen van toetsen. [Algorithms for constructing tests.] In W.J. van der Linden (Ed.), Moderne methoden voor toetsconstructie en -gebruik. Lisse: Swets & Zeitlinger.

Timminga, E. (1985). Geautomatiseerd toetsontwerp: Itemselectie m.b.v. binair programmeren [Automated test design: Item selection using binary programming]. Masters thesis, University of Twente, Enschede, Netherlands.

van der Linden, W.J. (1987). Automated test construction using minimax programming. In W.J. van der Linden (Ed.), IRT-based test construction (Research Report 87-2). Enschede: University of Twente, Department of Education.

van der Linden, W.J., & Boekkooi-Timminga, E. (1988). A zero-one programming approach to Gulliksen's matched random subtests method. Applied Psychological Measurement, 12, to appear.

Wagner, H.M. (1975). Principles of operations research. London: Prentice-Hall.

Yen, W.M. (1983). Use of the three-parameter model in the development of a standardized achievement test. In R.K. Hambleton (Ed.), Applications of item response theory. Vancouver: Educational Research Institute of British Columbia.

Authors' Note

The authors are indebted to Jos J. Adema for suggesting Equation 17 as a simplification of an earlier version of this constraint. This research was supported in part by a grant from the Dutch Organization for Advanced Research (ZWO) through the Foundation for Psychological and Psychonomic Research in the Netherlands (Psychon).

Table 1

CPU Times and Numbers of Iterations for Four Solutions

	CPU	Iterations
Optimal 0–1 solution (branch–and–bound)	4:11:42.30	32,570
Linear solution (simplex)	00:00:29.50	68
Optimal rounding (after simplex)	00:00:03.60	21
First 0–1 solution	00:01:13.60	153

Table 2

Description of the Solutions for the Four Approaches

	y	$I_t(-2)$	$I_t(0)$	$I_t(2)$	T	n
Optimal 0–1 solution	3.300	3.300	3.336	3.306	98	20
Rounded linear solution	3.187	3.374	3.187	3.273	102	20
Optimally rounded solution	3.300	3.300	3.336	3.306	98	20
First 0–1 solution	3.244	3.265	3.536	3.244	100	20

Table 3

Alternative Objective Functions in Test Construction

	Mathematical Formulation
1. Minimization of test length (Theunissen, 1985)	$\min \sum_{i=1}^{I} x_i$
2. Maximization of test information	$\max \sum_{k=1}^{K} \sum_{i=1}^{I} I_i(\theta_k)x_i$
3. Minimization of sum of positive deviations from the target (Timminga, 1985)	$\min \sum_{i=1}^{I} \sum_{k=1}^{K} I_i(\theta_k)x_i$
4. Minimization of sum of absolute deviation from the target	$\min \sum_{k=1}^{K} y_k + u_k$ subject to $\sum_{i=1}^{I} I_i(\theta_k)x_i - y_k + u_k = I(\theta_k)$, $k = 1, \ldots, K$ $y_k, u_k \geq 0$
5. Minimization of the largest positive deviation from the target where the target is met at all points (van der Linden, (1987)	$\min y$ subject to $\sum_{i=1}^{I} I_i(\theta_k)x_i - y \leq I(\theta_k)$, $k = 1, \ldots, K$ $y \geq 0$
6. Minimization of administration (Boekkooi-Timminga & van der Linden)	$\min \sum_{i=1}^{I} f_i x_i$
7. Minimization of frequencies of item administrations (Boekkooi-Timminga & van der Linden, 1987)	$\min \sum_{i=1}^{I} f_i x_i$

Titles of Recent Research Reports

RR-86-1 W.J. van der Linden, The use of test scores for classification decisions with threshold utility

RR-86-2 H. Kelderman, Item bias detection using the loglinear Rasch model: Observed and unobserved subgroups

RR-86-3 E. Boekkooi-Timminga, Simultaneous test construction by zero-one programming

RR-86-4 W.J. van der Linden, & E. Boekkooi-Timminga, A zero-one programming approach to Gulliksen's matched random subtests method

RR-86-5 E. van der Burg, J. de Leeuw, & R. Verdegaal, Homogeneity analysis with k sets of variables: An alternating least squares method with optimal scaling features

RR-86-6 W.J. van der Linden, & T.J.H.M. Eggen, An empirical Bayes approach to item banking

RR-86-7 E. Boekkooi-Timminga, Algorithms for the construction of parallel tests by zero-one programming

RR-86-8 T.J.H.M. Eggen, & W.J. van der Linden, The use of models for paired comparisons with ties

RR-86-9 H. Kelderman, Common item equation using the loglinear Rasch model

RR-86-10 W.J. van der Linden, & M.A. Zwarts, Some procedures for computerized ability testing

RR-87-1 R. Engelen, Semiparametric estimation in the Rasch model

RR-87-2 W.J. van der Linden (Ed.), IRT-based test construction

RR-87-3 R. Engelen, P. Thommassen, & W. Vervaat, Ignatov's theorem: A new and short proof

RR-87-4 E. van der Burg, & J. de Leeuw, Use of the multinomial jackknife and bootstrap in generalized nonlinear canonical correlation analysis

RR-87-5 H. Kelderman, Estimating a quasi-loglinear models for the Rasch table if the number of items is large

RR-87-6 R. Engelen, A review of different estimation procedures in the Rasch model

RR-87-7 D.L. Knol & J.M.F. ten Berge, Least-squares approximation of an improper by a proper correlation matrix using a semi-infinte convex program

RR-87-8 E. van der Burg & J. de Leeuw, Nonlinear canonical correlation analysis with k sets of variables

RR-87-9 W.J. van der Linden, Applications of decision theory to test-based decision making

RR-87-10 W.J. van der Linden & E. Boekkooi-Timminga, A maximin model for test design with practical constraints

Research Reports can be obtained at costs from Mediatheek, Department of Education, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands.

department of
EDUCATION

A publication by
the Department of Education
of the University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands